10 April 2017

Changes to Device Identifiers in Android O

Posted by Giles Hogben, Privacy Engineer

Android O introduces some improvements to help provide user control over the use of identifiers. These improvements include:

- limiting the use of device-scoped identifiers that are not resettable
- updating the Android O Wi-Fi stack in conjunction with changes to the Wi-Fi chipset firmware used by Pixel, Pixel XL and Nexus 5x phones to randomize MAC addresses in probe requests
- updating the way that applications request account information and providing more user-facing control

Device identifier changes

Here are some of the device identifier changes for Android O:

Android ID

In O, Android ID (Settings.Secure.ANDROID_ID or SSAID) has a different value for each app and each user on the device. Developers requiring a device-scoped identifier, should instead use a resettable identifier, such as Advertising ID, giving users more control. Advertising ID also provides a user-facing setting to limit ad tracking.

Additionally in Android O:

- The ANDROID_ID value won't change on package uninstall/reinstall, as long as the package name and signing key are the same. Apps can rely on this value to maintain state across reinstalls.
- If an app was installed on a device running an earlier version of Android, the Android ID remains the same when the device is updated to Android O, unless the app is uninstalled and reinstalled.
- The Android ID value only changes if the device is factory reset or if the signing key rotates between uninstall and reinstall events.
- This change is only required for device manufacturers shipping with Google Play services and Advertising ID. Other device manufacturers may provide an alternative resettable ID or continue to provide ANDROID ID.

Build.SERIAL

To be consistent with runtime permissions required for access to IMEI, use of android.os.Build.SERIAL is deprecated for apps that target Android O or newer. Instead, they can use a new Android O API, *Build.getSerial()*, which returns the actual serial number, as long as the caller holds the PHONE permission. In a future version of Android, apps targeting Android O will see Build.SERIAL as "UNKNOWN". To avoid breaking legacy app functionality, apps targeting prior versions of Android will continue see the device's serial number, as before.

Net.Hostname

Net.Hostname provides the network hostname of the device. In previous versions of Android, the default value of the network hostname and the value of the DHCP hostname option contained Settings.Secure.ANDROID_ID. In Android O, net.hostname is empty and the DHCP client no longer sends a hostname, following IETF RFC 7844 (anonymity profile).

Widevine ID

For new devices shipping with O, the Widevine Client ID returns a different value for each app package name and web origin (for web browser apps).

Unique system and settings properties

In addition to Build.SERIAL, there are other settings and system properties that aren't available in Android O. These include:

- **ro.runtime.firstboot**: Millisecond-precise timestamp of first boot after last wipe or most recent boot
- **htc.camera.sensor.front_SN**: Camera serial number (available on some HTC devices)
- **persist.service.bdroid.bdaddr**: Bluetooth MAC address property
- **Settings.Secure.bluetooth_address**: Device Bluetooth MAC address. In O, this is only available to apps holding the LOCAL_MAC_ADDRESS permission.

MAC address randomization in Wi-Fi probe requests

We collaborated with security researchers[1] to design robust MAC address randomization for Wi-Fi scan traffic produced by the chipset firmware in Google Pixel and Nexus 5X devices. The Android Connectivity team then worked with manufacturers to update the Wi-Fi chipset firmware used by these devices.

new random MAC address (whether or not the device is in standby).

- The initial packet sequence number for each scan is also randomized.
- Unnecessary Probe Request Information Elements have been removed: Information Elements are limited to the SSID and DS parameter sets.

Changes in the getAccounts API

In Android O and above, the GET_ACCOUNTS permission is no longer sufficient to gain access to the list of accounts registered on the device. Applications must use an API provided by the app managing the specific account type or the user must grant permission to access the account via an account chooser activity. For example, Gmail can access Google accounts registered on the device because Google owns the Gmail application, but the user would need to grant Gmail access to information about other accounts registered on the device.

Apps targeting Android O or later should either use AccountManager#newChooseAccountIntent() or an authenticator-specific method to gain access to an account. Applications with a lower target SDK can still use the current flow.

In Android O, apps can also use the AccountManager.setAccountVisibility()/ getVisibility() methods to manage visibility policies of accounts owned by those apps.

In addition, the LOGIN_ACCOUNTS_CHANGED_ACTION broadcast is deprecated, but still works in Android O. Applications should use addOnAccountsUpdatedListener() to get updates about accounts at runtime for a list of account types that they specify.

Check out Best Practices for Unique Identifiers for more information.

Notes

[1] Glenn Wilkinson and team at Sensepost, UK, Célestin Matte, Mathieu Cunche: University of Lyon, INSA-Lyon, CITI Lab, Inria Privatics, Mathy Vanhoef, KU Leuven ↩

Android O   Identity   Pixel

Newer post

Older post